

What is claimed is:

1. A storage medium for storing a compiler for compiling a source program, said compiler comprising the steps of:
  - detecting a parallelization directive in said source program; and
  - if said parallelization directive is detected, generating a front-end intermediate language for said parallelization directive by positioning on a storage region, each processing code of at least part of the parallelization directive with a hierarchical structure in accordance with an internal structure of said parallelization directive.
2. The storage medium according to claim 1, further comprising a step of adding to said front-end intermediate language of a statement to which the parallelization directive is applied, reference information from said front-end intermediate language of said statement to which the parallelization directive is applied, to said front-end intermediate language for the parallelization directive.
3. The storage medium according to claim 1, further comprising a step of, by using a processing table which stores one or a plurality of items of processing information for each of said processing codes, acquiring the processing information corresponding to a current processing content based on said processing code within the front-end intermediate language for said parallelization directive.
4. The storage medium according to claim 3, wherein said current processing content is one of type analysis, syntactic analysis, semantic analysis, and generation of a compiler intermediate language.
5. The storage medium according to claim 1, wherein said hierarchical structure is a list structure.
6. The storing medium according to claim 1, wherein
  - said part of said parallelization directive comprises a directive, a clause, and a line, and

a processing code for said directive is linked downward to a processing code for said clause, and

said processing code for said clause is linked downward to a processing code for said lines.

7. A compiling method for compiling a source program, said compiling method comprising the steps of:

detecting a parallelization directive in said source program; and

if said parallelization directive is detected, generating a front-end intermediate language for said parallelization directive by positioning on a storage region, each processing code of at least part of the parallelization directive with a hierarchical structure in accordance with an internal structure of said parallelization directive.

8. The compiling method according to claim 7, further comprising a step of adding to said front-end intermediate language of a statement to which the parallelization directive is applied, reference information from said front-end intermediate language of said statement to which the parallelization directive is applied, to said front-end intermediate language for the parallelization directive.

9. The compiling method according to claim 7, further comprising a step of, by using a processing table which stores one or a plurality of items of processing information for each of said processing codes, acquiring the processing information corresponding to a current processing content based on said processing code within the front-end intermediate language for said parallelization directive.

10. The compiling method according to claim 9, wherein said current processing content is one of type analysis, syntactic analysis, semantic analysis, and generation of a compiler intermediate language.

11. The compiling method according to claim 7, wherein said hierarchical structure is a list structure.

12. The compiling method according to claim 7, wherein  
said part of said parallelization directive comprises a directive, a clause,  
and a line, and  
a processing code for said directive is linked downward to a processing  
code for said clause, and  
said processing code for said clause is linked downward to a processing  
code for said lines.
13. A compiling apparatus for compiling a source program, comprising:  
means for detecting a parallelization directive in said source program;  
and  
means for generating a front-end intermediate language for said  
parallelization directive by positioning on a storage region, each processing  
code of at least part of the parallelization directive with a hierarchical  
structure in accordance with an internal structure of said parallelization  
directive if said parallelization directive is detected.
14. The compiling apparatus according to claim 13, further comprising  
means for adding to said front-end intermediate language of a statement to  
which the parallelization directive is applied, reference information from said  
front-end intermediate language of a statement to which the parallelization  
directive is applied, to said front-end intermediate language for the  
parallelization directive.
15. The compiling apparatus according to claim 13, further comprising  
means for, by using a processing table which stores one or a plurality of items  
of processing information for each of said processing codes, acquiring the  
processing information corresponding to a current processing content based on  
said processing code within the front-end intermediate language for said  
parallelization directive.
16. The compiling apparatus according to claim 15, wherein said current  
processing content is one of type analysis, syntactic analysis, semantic  
analysis, and generation of a compiler intermediate language.

17. The compiling apparatus according to claim 13, wherein said hierarchical structure is a list structure.

18. The compiling apparatus according to claim 13, wherein  
said part of said parallelization directive comprises a directive, a clause,  
and a line, and

a processing code for said directive is linked downward to a processing  
code for said clause, and

said processing code for said clause is linked downward to a processing code for  
said lines.